

AD-A222 266



# CLDGEN

## USERS GUIDE

by

Capt John A. Rupp

APRIL 1990

DTIC  
ELECTE  
MAY 24 1990

APPROVED FOR PUBLIC RELEASE;  
DISTRIBUTION IS UNLIMITED


USAF  
ENVIRONMENTAL TECHNICAL  
APPLICATIONS CENTER

Scott Air Force Base, Illinois, 62225-5438

90 05 24 07 8


## REVIEW AND APPROVAL STATEMENT

USAFETAC/TN-90/003, *CLDGEN Users Guide*, April 1990, has been reviewed and is approved for public release. There is no objection to unlimited distribution of this document to the public at large, or by the Defense Technical Information Center (DTIC) to the National Technical Information Service (NTIS).



PATRICK J. BREITLING  
Chief Scientist

FOR THE COMMANDER



WALTER S. BURGMAN  
Scientific and Technical Information  
Program Manager

## REPORT DOCUMENTATION PAGE

2. Report Date: April 1990
3. Report Type: Technical Note
4. Title: CLDGEN Users Guide
6. Author: Capt John A. Rupp
7. Performing Organization Name and Address: USAF Environmental Technical Applications Center (USAFETAC), Scott AFB, IL 62225-5438
8. Performing Organization Report Number: USAFETAC/TN-90/003
12. Distribution/Availability Statement: Approved for public release; distribution unlimited.
13. Abstract: Tells how to use USAFETAC's Cloud Scene Generator (CLDGEN) computer subroutine. Explains capabilities and limitations of the subroutine, with instructions for installing and running it on the user's computer system. Output from the subroutine is used by a user-supplied host program to tabulate statistics on cloud effects.
14. Subject Terms: CLIMATOLOGY, CLOUDS, CLOUD COVER, CLOUD SCENE, SIMULATION, MODEL, CLDGEN, SUBROUTINE.
15. Number of Pages: 17
17. Security Classification of Report: Unclassified
18. Security Classification of this Page: Unclassified
19. Security Classification of Abstract: Unclassified
20. Limitation of Abstract: UL

<b>Accession For</b>	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input checked="checked" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Available and/or	
Special	
A-1	

## PREFACE

The CLDGEN (Cloud Scene Generator) computer subroutine was developed by the United States Air Force Environmental Technical Applications Center's Environmental Simulation Branch (USAFETAC/DNY) for the Air Force Center for Studies and Analysis. CLDGEN was developed to perform cloud effects studies for Earth-based viewing environments. It is written in ANSI FORTRAN 77 and is designed to be installed on a host program.

## TABLE OF CONTENTS

INTRODUCTION.....	1
DESIGN PHILOSOPHY .....	2
SUBROUTINE CAPABILITIES AND LIMITATIONS .....	3
INTERACTION WITH SUBROUTINE .....	4
Subroutine Inputs.....	4
Coefficients .....	4
Initialization .....	4
Point Location .....	5
Subroutine Output.....	5
EXAMPLE APPLICATION.....	6
SUBROUTINE INSTALLATION.....	7
BIBLIOGRAPHY .....	8
APPENDIX A--Simple Application Algorithm.....	9
APPENDIX B--Installation Test Program .....	10

## INTRODUCTION

USAFETAC/DNY developed CLDGEN to perform sensitivity analyses of Earth-to-space satellite viewing through the atmosphere. The CLDGEN subroutine is designed to be included in a host program supplied by the user, who must write the routines that generate the statistics on the information required.

CLDGEN generates clouds at one point in the sky for each call to the subroutine. It can also generate lines, sectors, areas, or whole sky images as long as it is done pixel by pixel. The information from the model can be used to determine statistics on the amount of time a satellite is visible to an observer on the ground.

CLDGEN simulates clouds on two scales. First, it simulates representative sky cover over the entire celestial dome. Second, it simulates cloud/no cloud for an exact point in the sky. Individual cloud elements occur on a scale much smaller, both in space and time, than the scale representative of whole sky cover.

Seasonal and diurnal variations in cloud distribution over the ground site are accounted for by using data-compacted input variables. Mean sky cover and scale distance are used to define sky-cover distribution at each site. USAFETAC generates the site-specific data files.

## DESIGN PHILOSOPHY

CLDGEN is an all-purpose subroutine that can be used to assess the effects of cloud cover on an Earth-based viewing system. It combines techniques based on specialized random number generators, cloud-free line-of-sight (CFLOS) relationships, and modeled cloud-cover distributions. The subroutine is extremely flexible in that it accepts any possible combination of viewing geometry and time. CLDGEN was designed as a subroutine so that users have the widest flexibility in applying it to their specific needs. Users must create a separate driver program that passes the required information to the CLDGEN subroutine.

The first component of the simulator is the Burger Aerial Algorithm (Burger, 1985). This algorithm uses two coefficients for a given time period to reconstruct the entire sky-cover distribution. The CLDGEN subroutine also uses a functional relationship between total sky cover and the probability of CFLOS, known as the "Malick and Allen" (1979) algorithm. Together, these relationships allow the subroutine to determine the correct percent of occurrence of cloud cover for a

particular time and place. The next component of the model randomly generates a sample value from the distribution of all possible occurrences of cloud elements and cloud cover. A separate sawtooth generator (Gringorton and Boehm, 1984) is used to simulate the proper cloud-cover category, in tenths, and the distribution of clouds in the sky.

CLDGEN generates a value for sky coverage (in tenths) at a specified time. For some applications, this may be all the customer needs. However, the subroutine continues by also simulating the *distribution* of the various cloud elements. Given a user-specified point in the sky, the subroutine will also simulate cloud or no cloud. CLDGEN uses the sky coverage (simulated) and zenith angle (user-supplied) to calculate the probability of clouds based on the Malick and Allen algorithm. This probability value is then used as a threshold to determine the presence of cloud when compared to a random number. The second sawtooth generator is used to produce cloud elements with the same spatial scale as that of actual clouds.

## SUBROUTINE CAPABILITIES AND LIMITATIONS

This section discusses the capabilities and limitations of the CLDGEN subroutine. Its enhanced capabilities stem from the application of advanced techniques in simulation and sky cover climatology. The major advantage of the subroutine is its flexibility in addressing many Earth-based viewing problems. It should *not* be used for space-based viewing systems.

The CLDGEN subroutine is not a stand-alone program, and users must create routines that will provide access to it. CLDGEN's flexibility gives it a wide range of capabilities, such as the determination of clouds along arcs, within sectors, and whole-sky images. Although the subroutine can accommodate any combination of viewing geometry, it can compute cloud values for only one point at a time. To get information for any line or area in the sky, users must access the subroutine for each individual point that makes up the line or area.

The subroutine can handle any combination of multiple satellites at different times, as long as it receives one point at a time. The position and time of each satellite must be passed to CLDGEN separately.

CLDGEN is extremely flexible due to the application of sawtooth generators. One advantage of sawtooth generators is that they generate random variables for any desired time. There is no need to use a fixed time step to jump ahead to the desired time. This is a clear advantage over other simulation techniques that require equally spaced time steps. For example, a user may want cloud information for every minute during 1 July, and then jump directly to 30 December. The sawtooth model accounts for the cloud correlation time. It also simulates cloud elements that have the same spatial correlation as that of actual clouds.

Another advantage of CLDGEN is that a weather sequence can be repeated. This allows users of complicated war gaming simulation to replay the same weather sequence, if desired. Users can test the other components of a simulation without having an external factor like weather change. For example, even if weather will affect a weapon system adversely most of the time, the user may be interested in testing changes in the weapons system's operations due to changes in non-weather factors, such as logistics, sequencing, and manpower.

CLDGEN uses a set of modeling coefficients that describe the distribution of cloud cover. It uses these coefficients to tune the model so that it will reproduce cloud statistics that will match real-world sky-cover distributions. However, locally induced features caused by terrain or geography are not modeled. For example, a region of preferred cloudiness near a mountain top or coastal area will not be reproduced by the subroutine. CLDGEN also assumes that all clouds occur at a single level. It allows no multiple cloud layers and no distinction between cloud types. The user will not be able to tell if a cloud is a stratocumulus or cirrus.

Another limitation is that CLDGEN is designed to give information for a single site. Once primed, it will simulate variables that are representative for a single site based on the input coefficients. For example, if the model is initialized and given coefficients for White Sands, NM, users cannot "trick" it into simulating cloud data for Eglin AFB, FL. To simulate for a different site, the model must be re-initialized and given coefficients for the new site. Although it is possible for users to run a single simulation that will simulate clouds at several stations simultaneously, USAFETAC/DNY should be contacted for more information on such applications.



## INTERACTION WITH SUBROUTINE

**INTERACTION.** This section shows users how to interact with the CLDGEN subroutine, which is written in FORTRAN 77 and uses ANSI FORTRAN conventions. Users should be familiar with these conventions when working with CLDGEN. Subroutine inputs and outputs are addressed separately.

**SUBROUTINE INPUTS.** CLDGEN has three types of inputs: coefficient inputs, initialization inputs, and point location inputs.

**Coefficients.** First, users must obtain data for the locations of interest. All requests for such data are directed to USAFETAC IAW AWSR 105-18.

Coefficient inputs for the subroutine are accessed through an external file. Users must ensure that the program has access to that file, which contains a set of coefficients used to reconstruct the distribution of total sky cover. These two coefficients are known as mean sky cover and scale distance, and are produced from climatological data. For a more complete description of these modeling coefficients, see Burger (1985).

The coefficient dataset has a name identifier and 48 rows of coefficients. There are 12 columns across that correspond with the month. The format of the file is illustrated in Table 1.

**TABLE 1. Input Coefficient File Format.**

<u>VARIABLE</u> SNAM	<u>COLUMN</u> 1-4	<u>FORMAT</u> A4	<u>DESCRIPTION</u> site I.D
for K=1,24 hour of day (GMT)			
XM(month,K)	1-85	1X,12F7.3	mean sky cover
SDIS(month,K)	1-85	1X,12F7.3	scale distance

**Initialization.** At the heart of all simulation models is the process that generates random numbers. The random number generator is responsible for producing a random variable with the desired statistical characteristics. In this context, the numbers generated will appear as independent draws from some given distribution. A simulation model may have more than one type of random number generator. Some random numbers are used to set up or "initialize" other random number generators. This is the case with the "sawtooth"

generator used at USAFETAC. During the initialization stage of a sawtooth subroutine, numbers are selected from the random number sequence, beginning at the designated starting point. This number sequence will be identical for every run of the simulation. Because of this unique quality, independent simulations will result from choosing a different starting point in the sequence. Table 2 illustrates two different starting points in a same random number sequence.

**TABLE 2. Sample of 2 NCALLS Random Number Sequences.**

<u>Run #1</u>	<u>Run #2</u>
.454 start	.454
.934	.934
.275	.275
.537	.537 start
.345	.345
.914	.914
.563	.563
.699	.699
.223	.223
.	.
.	.

Users must select the starting point in the sequence. This value, called "NCALLS," is an integer showing the *i*'th point in the sequence. The CLDGEN subroutine uses the NCALLS value only during the initialization stage and is indicated when the variable "PRIME" is set to 1.0. The subroutine should only be initialized at the beginning of the subroutine run; if not, it will read the same input twice and go past the end of the file.

Users must select a different value of NCALLS to create an independent run. If the same NCALLS is used each time CLDGEN is primed, the same sequence of weather will be repeated, assuming that all other input variables are the same. Users can go back to the same time and look angles, and encounter the same weather.

**Point Location.** CLDGEN requires that the user supply the location in the sky and the time of interest. Viewing geometry is required in two angular measurements; these two inputs are the azimuth angle in degrees (AZANG), and the zenith angle in degrees (ZENANG). This information defines the exact location in the sky that the ground observer is looking at. The next variable to be supplied by the user is the Julian time in hours (JULTIM), or the number of hours (GMT) that have elapsed since the beginning of the year. Fractional hours can be used to indicate time periods of less than an hour.

For example, on 10 January at 12Z, the Julian time has a value of 228.00. At 1230Z on the same date, JULTIM is 228.50.

Once the subroutine has been initialized in a sawtooth model, the sequence of weather is completely determined, allowing the user to turn the clock back and pick a new point in the sky. The user can follow one satellite, observing how clouds interfere with it, then look at the second satellite by turning the clock backwards or forwards. As long as the proper information on the location and time is passed to CLDGEN, it is possible to observe any number of satellites.

**SUBROUTINE OUTPUT.** The CLDGEN subroutine produces four output variables for every call to the subroutine; they are: "CLOUD", "IBIN", "BINEND" and "CLDEND." CLOUD indicates the presence of a cloud for the user's specified time and location. A value of 1 is cloudy, and 0 is clear. IBIN indicates the category of total sky cover. As shown in Figure 1, there are 11 categories. The first, IBIN=1, represents sky cover from 0 to 5 percent. The last, IBIN=11, is sky-cover from 95 to 100 percent. All other categories, (2 thru 10) have a width of 10 percent.

sky-cover category (IBIN)	1	2	3	4	5	6	7	8	9	10	11
sky-cover percentile	0-5	5-15	15-25	25-35	35-45	45-55	55-65	65-75	75-85	85-95	95-100

*Figure 1. Categories of sky-cover distribution and associated percentiles.*

A minor difficulty in working with meteorological variables is that they are not usually normally distributed. A technique called "transnormalization" (Boehm, 1976) can be used to transform a non-normal distribution to a normal distribution. This technique is used in CLDGEN. Transnormalization produces a value known as an "Equivalent Normal Deviate," or END. The END gives the number of standard deviations from the mean of a normal distribution. The END value is then compared to the "normalized" distribution of cloud cover to determine the cloud-cover category. Because the subroutine

simulates two variables, there is an END generated for BINEND and CLDEND. BINEND is the END for the sky cover category IBIN. CLDEND is the END for occurrence of cloud. BINEND and CLDEND are diagnostic variables used in the process of verifying the correct installation of the subroutine. They are not useful to the user except in testing the subroutine. For a discussion on the use of BINEND and CLDEND for testing the subroutine, see "Subroutine Installation" on page 7.

## EXAMPLE APPLICATION

This example walks the user through the steps required for using the CLDGEN subroutine. Note that since an infinite number of study combinations for measuring cloud effects is possible, users must take care in creating the type of study that will answer their unique questions.

Suppose that a user wants to determine average cloudiness along an arc in the sky over a period of a year. The arc extends from a point due west (270 degrees) at an elevation of 10 degrees (zenith angle = 80) to a point directly overhead. First, the user must determine the set of points that will give a representative sample of cloudiness along the arc. A sample at every degree, which gives a total of 81 points for the arc, should provide adequate resolution for most applications. The next step is to determine the number of times that the arc will be sampled. An arbitrary time step of 1 hour is selected so that a sample is taken at every hour of every month, providing a loop through all of the hours in the year. For every hour, there will be a sample of 81 points in the sky. A summation of the number of cloudy pixels, compared to all pixels, gives us an estimate of the average cloudiness along the arc for a given simulation run. (A "pixel" represents a narrow pencil beam making a single point in the sky.) This experiment can be repeated to get a composite average. Once we have determined these steps, a FORTRAN driver program can be written to access the CLDGEN subroutine for simulated cloud data.

We will now take the description of the problem as presented in the previous paragraph and write a sample shell of a FORTRAN driver (Appendix A). The algorithm shown in Appendix A is incomplete in that it shows only the major steps of the example. Users must fill in their own system-specific items such as dimensioning arrays and writing the output statements.

**STEP #1:** The first step in the program is to initialize the angles and counters by setting them to 0. Here, the viewing angles are also set to zero. The counters, SUMCLR and SUMCLD, will be used to hold the sum of clear and cloudy points, respectively.

**STEP #2:** At the beginning of every simulation, the subroutine must be primed. If it is not primed, the

program will give incorrect results from a division by zero error, and the program will probably be terminated abnormally. During the initialization stage, the input variable PRIME is set to a value of "1." When the subroutine gets the signal to prime, it uses the NCALLS value; this example arbitrarily uses 25. In order for a simulation run to be independent of other simulation subroutine runs, it must have a unique value for NCALLS.

**STEP #3:** The subroutine is called once at the beginning to perform the initialization.

**STEP #4:** After initialization, the value PRIME is set to "0." This tells the subroutine that we will not be initializing.

**STEP #5:** The program will make several loops to cover all the times and points directed by the user's requirements. The outer loop counts through the value of I HOUR for every hour of the year. The value JULTIM is calculated by converting the integer value of I HOUR into a real number. Next, there is a loop counting from 0 to 80. This is the integer value of the zenith angle. The value of ZENANG is then computed by converting the integer value of "J" into a real number. Next, the CLDGEN subroutine is called to get the cloud information for the specified time and point.

**STEP #6:** The values of clear or cloudy are summed over the entire length of the simulation year. The value "CLOUD" will tell us if it is cloudy at the point specified. When CLOUD is equal to 1, that point is considered cloudy. If CLOUD equals 0, it is clear. The values of SUMCLR and SUMCLD hold the sum of all clear and cloudy pixels, respectively.

**STEP #7:** The total value of percent cloudiness over the arc is calculated by dividing the cloudy pixels by the total number of pixels.

We have used a very simple example here. Most users, of course, will have much more complicated scenarios, and it is left to them to decide how information on the presence of clouds--the only thing provided by this simulation subroutine--is used.

## SUBROUTINE INSTALLATION

The following three steps are required to install and test the CLDGEN subroutine.

### STEP #1 Load, Compile, and Link.

The first step in this process is to have the subroutine loaded on the system. The subroutine is shipped on tape as source code. The source code, along with the user driver program, must then be compiled and link edited. The CLDGEN subroutine, with the test driver and test dataset, is in the following format:

```
ASCII  
6250 BPI  
Record length 132  
Block size 3960  
Unlabeled
```

### STEP #2 Correct random number generator.

The second step is to convert the random number generator in the program to be functional for the user's system. The subroutine in CLDGEN that creates random numbers is specific to the number of bits in an integer word. This varies based on your computer system. The random number generator furnished with CLDGEN (RANDW) is for a computer with a 32-bit integer word. If your system uses a word other than 32 bits, you must correct the scaling factor in function RANDW. In the RANDW function, the scaling factor adjusts the random number so that it is within the correct range between zero and one. The correction factor is  $1.0/(2.0^{**}(N-1))$ , where N is the number of bits in an integer word on the computer being used. The user must adjust the RANDW function to account for the correct bit number of the user's system.

CLDGEN comes with a temporary function called SIMRAN, which takes the place of RANDW only to ensure the correct transportation of the subroutine. SIMRAN is used as a predictable but inaccurate random number generator which will run on any system regardless of integer bit numbers. By using the code as delivered with the SIMRAN function, the user can tell if the subroutine is at least functioning before installing the correct RANDW function. All references to the RANDW function have been mentioned in the original version of CLDGEN. Once the subroutine has been placed on the user's system, all references to SIMRAN should be changed over to RANDW. The SIMRAN function should be discarded once the user has installed the program.

### STEP #3 Test for Accuracy.

After the first two steps are complete and CLDGEN is installed correctly, the user should run a test algorithm (supplied in Appendix B) to ensure the subroutine is working correctly. The test algorithm will perform diagnostic checks of the performance of the simulation subroutine. The routine will calculate the mean and standard deviation of the variables CLDEND and BINEND. When calculating means and standard deviations, the user should go through at least 100,000 repetitions so as to produce a very long record of data. A large sample size is needed to ensure that the statistics produced are stable. For both CLDEND and BINEND, the resulting means should be very close to zero (0.0) ( $\pm .03$ ) and the standard deviation should be very close to one (1.0) ( $\pm .05$ ). The test program can be used to print the means and standard deviations to the output unit. If these means and standard deviations vary significantly from 0 and 1, respectively, the subroutine should be considered suspect; contact USAFETAC/DNY for assistance.

## BIBLIOGRAPHY

Boehm, A.R., *Transnormalization Regression Probabilities*, AWS-TR-75-259, Air Weather Service, Scott AFB, IL, 1976.

Burner, C.F., "A New Model for Specifying Cloud Distributions for Climatologies," *Preprints, Ninth Conference on Probability and Statistics in Atmospheric Sciences*, American Meteorological Society, pp. 459-462, 1985.

Gringorton, I.I., and A.R. Boehm, *The 3D-BSW Model Applied to Climatology of Small Areas and Lines*, AFGL/TR-87/0251, Air Force Geophysics Laboratory, Hanscom AFB, MA, 1987

Malick, J.D., J.H. Allen, and S. Zakanycz, *Calibrated Analytical Modeling of Cloud-Free Intervals*, SPIE, 195, pp. 142-147, 1979.

## Appendix A

### SIMPLE APPLICATION ALGORITHM

```
C STEP #1 Set counters to zero and set all angles to zero except set
C azimuth angle to 270.

ZENANG=0.
AZANG=270.
SUMCLR=0.
SUMCLD=0.
C
C STEP #2 Turn on priming flag and set NCALLS value
C
PRIME=1
NCALLS=25
C
C STEP #3 Prime model by calling CLDGEN subroutine
C
CALL CLDGEN (PRIME,NCALLS,ZENANG,AZANG,JULTIM,CLOUD,IBIN,BINEND,CLDEND)
C
C STEP #4 turn off priming flag
C
PRIME=0
C
C STEP #5 Make loops through all hours and zenith angles
C and keep a count of the cloudy and clear occurrences.
C
DO 20 IHOURL=1,8760
JULTIME=REAL(IHOURL)
DO 10 J=0,80
ZENANG=REAL(J)
CALL CLDGEN(PRIME,NCALLS,ZENANG,AZANG,JULTIM,CLOUD,
* IBIN,BINEND,CLDEND)
C
C STEP #6 Sum occurrences of clear and cloudy
IF(CLOUD.EQ.1) THEN
SUMCLD=SUMCLD +1.
ELSE
SUMCLR=SUMCLR +1.
ENDIF
10 CONTINUE
20 CONTINUE
C
C STEP #7 Calculate the total percentage of cloudiness
TOTAL=SUMCLD/(SUMCLD+ SUMCLR)
C
END
```

## Appendix B

### INSTALLATION TEST PROGRAM

A sample method to calculate the standard deviation, variance and mean of two variables CLDEND and BINEND.

C STEP #1 Set all counters to zero.

```
BINA=0.  
BINB=0.  
CLDA=0.  
CLDB=0.  
NCOUNT=0
```

C

C STEP #2 Turn on priming flag and set NCALLS value

C

```
PRIME=1  
NCALLS=25
```

C

C STEP #3 Prime model by calling CLDGEN subroutine

C

```
CALL CLDGEN(PRIME,NCALLS,ZENANG,AZMANG,JULTIM,CLOUD,IBIN,BINEND,CLDEND)
```

C

C STEP #4 Turn off priming flag

C

```
PRIME=0
```

C

C STEP #5 Make loops through at least 100000 hours and accumulate the values  
C to calculate the variance of BINEND and CLDEND

C

```
DO 20 IHOUR=1,100000  
  JULTIME=REAL(IHOUR)  
  CALL CLDGEN(PRIME,NCALLS,ZENANG,AZANG,JULTIM,CLOUD,IBIN,  
  *      BINEND,CLDEND)
```

C

```
BINA=BINA + BINEND  
BINB=BINB + BINEND**2  
CLDA=CLDA + CLDEND  
CLDB=CLDB + CLDEND**2  
NCOUNT=NCOUNT + 1
```

20 CONTINUE

C

C STEP #6 Calculate the standard deviation (BSTD) and the mean for variable  
C BINEND.

```
BVAR=(BINB-((BINA**2)/REAL(NCOUNT)))/(REAL(NCOUNT)-1.)  
BSTD=SQRT(BVAR)  
BMEAN=BINA/REAL(NCOUNT)
```

--Continued on next page--

C

C STEP # 7 Calculate the standard deviation (CSTD) and mean of the variable CLDEND.

CVAR=(CLDB-((CLDA\*\*2)/REAL(NCOUNT)))/(REAL(NCOUNT)-1.)

CSTD=SQRT(CVAR)

CMEAN=CLDA/REAL(NCOUNT)

C STEP #8 Write output variables to diskfile or video.

END



## DISTRIBUTION

AWS/DO, Scott AFB, IL 62225-5008 .....	1
AWS/XT, Scott AFB, IL 62225-5008 .....	1
AWS/XTX, Scott AFB, IL 62225008 .....	1
AWS/PM, Scott AFB, IL 62225-5008 .....	1
OL A, HQ AWS, Buckley ANG Base, Aurora, CO 80011-9599 .....	1
AFOTEC/WE, Kirtland AFB, NM 87117-7001 .....	1
CACDA, OL-E, HQ AWS, ATZL-CAW-E, Ft Leavenworth, KS 66027-5300 .....	1
SD/CWDA, PO Box 92960, Los Angeles, CA 90009-2960 .....	1
OL-H, HQ AWS (ATSI-CD-SW), Ft Huachuca, AZ 85613-7000 .....	1
OL-I, HQ AWS (ATWE), Ft Monroe, VA 23651-5051 .....	1
OL-K, HQ AWS, NEXRAD Opnl Facility, 1200 Westheimer Dr. Norman, OK 73069 .....	1
OL-L, HQ AWS, Keesler AFB, MS 39534-5000 .....	1
OL-M, HQ AWS, McClellan AFB, CA 95652-5609 .....	1
Det 1, HQ AWS, Pentagon, Washington, DC 20330-6560 .....	1
Det 2, HQ AWS, Pentagon, Washington, DC 20330-5054 .....	1
Det 3, HQ AWS, PO Box 3430, Onizuka AFB, CA 94088-3430 .....	1
1WW/DN, Hickam AFB, HI 96853-5000 .....	1
2WW/DN, APO New York 09094-5000 .....	1
3WW/DN, Offutt AFB, NE 68113-5000 .....	1
4WW/DN, Peterson AFB, CO 80914-5000 .....	1
2WS/DON, Andrews AFB, MD 20334-5000 .....	19
5WW/DN, Langley AFB, VA 23665-5000 .....	1
AFGWC/SDSL, Offutt AFB, NE 68113-5000 .....	3
USAFETAC, Scott AFB, IL 62225-5438 .....	6
7WW/DN, Scott AFB, IL 62225-5008 .....	1
3350 TECH TG/TTGU-W, Stop 62, Chanute AFB, IL 61868-5000 .....	2
WL/WE, Kirtland AFB, NM 87117-7001 .....	1
AFCSA/SAGW, Washington, DC 20330-5000 .....	1
NAVOCEANCOMFAC, Stennis Space Ctr, MS 39529-5002 .....	1
COMNAVOCEANCOM, Code N312, Stennis Space Ctr, MS 39529-5000 .....	1
NAVOCEANO, Code 9220 (Tony Ortolano), Stennis Space Ctr, MS 39529-5001 .....	1
FLENUMOCEANCEN, Monterey, CA 93943-5006 .....	1
NERL, Monterey, CA 93943-5006 .....	1
Naval Research Laboratory, Code 4323, Washington, DC 20375 .....	1
Naval Postgraduate School, Chmn, Dept of Meteorology, Code 63, Monterey, CA 93943-5000 .....	1
Naval Oceanography Command Ctr, COMNAVMAR Box 12, FPO San Francisco, CA 96630-5000 .....	1
Pacific Missile Test Center, Geophysics Division, Code 3253, Pt Mugu, CA 93042-5000 .....	1
Dept of Commerce/NOAA/MASC, Library MC5 (Jean Bankhead), 325 Broadway, Boulder, CO 80303 .....	2
Federal Coordinator for Meteorology, Suite 300, 11426 Rockville Pike, Rockville, MD 20852 .....	1
NOAA Library-EOC4WSC4, Attn: ACQ, 6009 Executive Blvd, Rockville MD 20852 .....	1
NOAA/NESDIS (Attn: Capt Taylor), FB #4, Rm 0308, Suitland, MD 20746 .....	1
GL Library, Attn: SULLR, Stop 29, Hanscom AFB, MA 01731-5000 .....	1
Atmospheric Sciences Laboratory, Attn: SLCAS-AT-AB, Aberdeen Proving Grounds, MD 21005-5001 .....	1
Atmospheric Sciences Laboratory, White Sands Missile Range, NM 88002-5501 .....	1
U.S. Army Missile Command, ATTN: AMSMI-RD-TE-T, Redstone Arsenal, AL 35898-5250 .....	1
Technical Library, Dugway Proving Ground, Dugway, UT 84022-5000 .....	1
NCDC Library (D542X2), Federal Building, Asheville, NC 28801-2723 .....	1
DTIC-FDAC, Cameron Station, Alexandria, VA 22304-6145 .....	2
AUL/LSE, Maxwell AFB, AL 36112-5564 .....	1
AWSTL, Scott AFB, IL 62225-5438 .....	50